УДК 631.3.07

Олег Клічук
Чернівецький національний університет імені Юрія Федьковича, Україна

Обробка реляційних баз даних засобами функціонального програмування

У статті розглядається методика використання функціонального програмування. Описано задачі, у яких використовуються функції, застосування яких суттєво спрощує проведення аналізу даних.

Питання обробки інформації у реляційних базах даних є одним із тих, які на сучасному етапі розвитку інформаційних технологій набули широкого застосування. Вони мають поліфонічний вплив як на практичний, так і на теоретичний напрямки наукового пошуку. Ця тема в основному висвітлена у закордонному науковому здобутку [1, с. 15], [2, с. 33]. У вітчизняній науковій думці питання використання функціонального програмування не набуло достатньої утилітарної пропозиції, хоча є численна кількість теоретичної інформації про перспективи їх практичного використання. Взагалі така проблематика є порівняно новою галуззю в інформаційних системах, тому й перед автором постає завдання провести послідовний аналіз теоретичних пропозицій з використання функціонального програмування, які у майбутньому можуть принести позитивний практичний результат та мати широке застосування. З цієї же причини може виникати ряд складнощів у розкритті питання, що для такої тематики є природним явищем. Дослідження нового викликає і скептичний погляд, і зауваження, і суперечності у питаннях герменевтики. Такі нюанси передбачаються попри все, тому вважаємо висвітлення заявленої теми доцільним та актуальним у вітчизняній науці, оскільки активне використання реляційних баз даних є очевидним та практично значущим [3, с. 102].

Функціональні мови використовуються для дослідження семантики двояко. Один спосіб – це опис інтерпретатора для мови, яка вивчається; функціональні мови ідеально пристосовані для цієї ролі. Другий спосіб полягає у тому, щоб для кожної програми на вибраній мові визначити еквівалентні функції або функціональну програму. У будь-якому випадку простота і потужність функціональної мови роблять її дуже зручною для таких семантичних специфікацій.

Функціональне програмування – це спосіб складання програм, в яких єдиною дією є виклик функції, єдиним способом розділення програми на частини є задання імені для функції і вказування для цього імені виразу, який обчислює значення функції, а єдиним правилом композиції – оператор суперпозиції функцій.

Функціональний стиль програмування базується на використанні тільки процедур-функцій. Роль змінних виконують параметри функцій, присвоєння значень здійснюється тільки при заданні аргументів у зверненнях до функцій, послідовна композиція операторів замінюється почерговим обчисленням аргументів при виклику функції, умовна композиція операторів – такою же композицією виразів, циклічна композиція – рекурсією.

На перший погляд може здатися, що арсенал засобів програмування надто збіднюється. Але це не так. Програми, які написані у функціональному стилі, звичайно є невеликі, наглядні і зручніші для розуміння. До того ж вони є надійнішими – «змінна»

(новий екземпляр параметра функції) створюється у момент присвоєння їй значення і не змінює його весь час свого існування. Використання рекурсії при виклику функції добре поєднується з рекурсивним стилем опису типів значень і таких конструкцій, як відображення – звертань до функції (або навпаки), тому що вираз – це не що інше, як розгалужена композиція звертання до вбудованих у мову функцій – операцій.

Звичайно, у кожної медалі є зворотній бік. Якщо список полів у типі запису дуже подібний на список параметрів процедури, то тип масиву має скоріше послідовний, ніж паралельний характер об'єднання компонент у складене значення. Обробка масивів гармонує з використанням циклів з параметром, але останні помітно гірше поєднуються з рекурсією, ніж цикли з перед- або післяперевіркою.

У роботах зі штучного інтелекту доводиться оперувати складними символьними структурами даних і такими ж складними алгоритмами. Тому використання чисто функціонального стилю програмування підходить для додатків, які характеризуються багатократним отриманням складних структур даних з інших таких же структур.

Ще однією проблемою при використанні функціонального програмування є задання множини наперед визначених вбудованих функцій, або примітивів. Вони використовуються для виконання простих операцій, аналогічно базовим арифметичним, можна будувати нові функції, для виконання більш складних операцій за допомогою цих примітивів. Також такі нові функції можна використовувати як блоки для створення більш складних функцій і так далі.

Як і в традиційній практиці програмування, ми можемо визначити нові функції, щоб або розділити і, відповідно, спростити визначення більш складної функції, або описати стандартну операцію і не переписувати однаковий вираз багатократно.

Фундаментальною властивістю математичних функцій, яка дозволяє зібрати разом інші функції, є функціональність (прозорість у посиланнях). Існує декілька інтуїтивних тверджень цього терміна, але по суті він означає, що кожен вираз визначає єдину величину, яку не можна змінювати ні шляхом її обчислення, ні наданням різним частинам програми можливості сумісно використовувати цей вираз. Обчислення виразу просто змінює форму виразу, але не змінює його величину. Всі посилання на деяку величину еквівалентні самій цій величині, і той факт, що на вираз можна посилатися з іншої частини програми, ніяк не впливає на величину цього виразу. Функціональність (прозорість у посиланнях) визначає різницю між математичними функціями і функціями, які можна написати на імперативних мовах програмування, таких, як Паскаль, оскільки ці мови дають функціям можливість посилатися на глобальні дані і дозволяти використовувати присвоєння, що може привести до зміни значення функції при повторному її виклику. Такі динамічні зміни у величині даних часто називаються побічними ефектами. Через них значення функції може змінюватися, навіть якщо її аргументи і залишаються без зміни кожного разу, коли до неї звертаються. Це приводить до того, що функцію важко використовувати, оскільки для того, щоб визначити, яка величина буде отримана при обчисленні функції, необхідно розглянути текучу величину глобальних даних. Це, у свою чергу, вимагає розгляду історії обчислень для визначення того, що породжує величину глобальних даних в кожен момент часу.

Вибір початкових функцій залежить від вибраних методів обробки даних.

Такі особливості функціонального програмування можна застосувати для обробки реляційних баз даних.

Реалізацію функціонального програмування у системах управління базами даних можна використати у таких задачах:

– проведення групування засобами функцій;

– проведення матричних обчислень;
– проведення пошуку параметрів заміною даних;
– проведення «нечітких» обчислень;
– реалізація тригерів;
– проведення обчислень по полях, які не належать одному запису.

Вибір сукупності основних функцій, безумовно, дуже важливий для практичних цілей: як для достатньо ефективної реалізації на сучасних машинах, так і для легкості відображення задачі програмою набір основних функцій повинен бути достатньо потужним. Сучасні системи управління базами даних містять великий набір вбудованих функцій, які можна використати для реалізації поставленої задачі. Як правило, мовою програмування є система Паскаль або аналогічні їй.

Відомо, що технологія обробки інформації нерозривно пов’язана з термінами бази даних, системи управління базами даних. Після створення і розвитку системи управління базами даних (СУБД) виникла ілюзія, що всі проблеми інформаційного забезпечення розв’язані. СУБД стала причиною виникнення нових проблем. Ці проблеми включають протиріччя між складністю і недостатньою ефективністю універсальних СУБД і обмеженими можливостями спеціалізованих систем. Для підвищення наглядності створених додатків у сучасних системах управління базами даних є можливість проводити обробку декількома етапами, тобто присвоєнням полям запитів проміжних обчислень.

Тому, не дивлячись на ситуацію, яка є сьогодні у програмуванні, продовжує існувати тенденція, що направлена на забезпечення все більш абстрактних шляхів вирішення проблеми, при цьому жертвують швидкістю обчислення програми для забезпечення простоти програмування. Такий підхід можна реалізувати у реляційних базах, не перетворюючи дані у різноманітні структури.

Використання інтелектуальних засобів обробки інформації у базах даних вимагає нових методів проведення обчислень. Засоби SQL-запитів, які передбачають проведення лише арифметичних обчислень по записах та підсумкових по групах записів, не забезпечують сучасні методи обробки даних, серед яких є групування даних за різними критеріями. Проведення групування засобами функцій забезпечується створенням функцій, які будують сукупності записів за деякою ознакою. Потужними засобами кластеризації володіють методи нейронних мереж, «нечітка» логіка та інші. Ці методики можна легко реалізувати функціональним програмуванням.

Інтерфейс збереження даних у вигляді двовимірних таблиць забезпечує наглядність інформації та проведення множинних операцій. Але операції традиційного обчислення на основі послідовного виконання інструкцій неможливо провести, так як обробка проводиться послідовно по записах відношень.

Характерною рисою функціонального програмування є той факт, що, незважаючи на відсутність заданого порядку обчислень, результат визначений однозначно, іншими словами: значення виразу (функції) – це величина і задача комп’ютера спростити вираз і обчислити його значення. У відношеннях значення полів записів є змінними функції, а поле, яке містить функцію, є її результатом обчислень.

Важливим елементом функціонального програмування є побудова композицій функцій. Функціональна програма складається із сукупності визначень функцій. Функції, у свою чергу, являють собою виклик інших функцій і засобів, які керують послідовністю викликів, і так далі відповідно до ієрархії визначень. Функції часто або прямо, або опосередковано викликають самі себе (рекурсія).

Кожен виклик повертає деяке значення у функцію, яка його викликала, цей процес повторюється до тих пір, поки початкова функція не поверне кінцевий результат користувачу. Такі методи обчислень зустрічаються при роботі нейронних мереж зі зворотніми зв'язками.

Такі обчислення у реляційних базах даних можна реалізувати створенням проміжних таблиць, значення з яких після кожної процедури виклику функції самої себе заноситься у початкову таблицю командами обновлення даних.

Можливості функціонального програмування можна застосувати для моделювання операцій над нечіткими даними. Такий підхід забезпечує зручність, наглядність та швидку реалізацію засобами систем управління базами даних.

Нечіткі дані являють собою звичайне узагальнення важливого математичного і логічного поняття звичайних множин на випадок, коли степінь приналежності елементів цих множин не є обов'язково повною або абсолютною. Тобто нечіткість виникає тоді, коли приналежність елементів до множин має тільки частковий характер, при цьому міра цієї частковості може мати як об'єктивний характер, обумовлений зовнішніми обставинами, так і визначатися суб'єктивно.

Проведення «нечітких» обчислень передбачає використання лінгвістичної змінної.

Це набір, до складу якого входять:

– назва лінгвістичної змінної;
– універсальна множина X, або область визначення лінгвістичної змінної;
– початкові значень базових терм-множин;
– процедура генерації нових лінгвістичних термів.

Основними задачами, які можна розв'язувати засобами функціонального програмування, є:

– задання функції приналежності;
– знаходження об'єднання множин нечітких даних;
– знаходження перерізу множин нечітких даних;
– знаходження доповнення до множини нечітких даних;
– реалізація операції концентрації та розмиття множини нечітких даних;
– знаходження множини рівня нечітких даних;
– знаходження декартового добутку.

Для реалізації таких обчислень зручно створити функції, які обчислюють значення приналежностей відповідно до початкових даних. Після того побудувати запити, які відображають результати обчислень. На основі запитів, які відображають початкові, базові терми користувача можна побудувати інші на основі множинних операцій. Як правило, такі запити використовують операцію «мінімум» для перерізу множин та «максимум» – для об'єднання. Для зміни параметрів функцій і відповідного моделювання можна використовувати параметричні запити.

Досить важливим моментом застосування нечітких даних є вибір моделі проведення обчислень з ними. У даний час відсутня єдина думка про те, як визначати операції перерізу та об'єднання нечітких множин, і існують різні варіанти означень.

Л.А. Заде вперше запропонував означення, які аксіомно визначають властивості операцій з нечіткими даними:

– перерізом двох нечітких множин A і B є найбільша нечітка множина, яка міститься і в A, і в B одночасно;
– об'єднанням A і B є найменша нечітка множина, яка містить хоча б один елемент з нечітких множин A або B.

Єдиними операціями, які задовольняють ці аксіоми, є наступні:

– переріз нечітких множин:

$$\mu C(x) = \min(\mu A(x), \mu B(x)), (C = A \cap B),$$

– об’єднання нечітких множин:

$$\mu D(x) = \max(\mu A(x), \mu B(x)), (D = A \bigcup B).$$

Введені таким чином математичні операції мають «жорсткий» характер, тобто при перерізі відсутня можливість компенсації значень, які належать μA(x), якими-небудь значеннями μB(x) і навпаки.

Тому можна використовувати більш «м’які» означення операцій перерізу і об’єднання, які за формою подібні до класичних операцій з ймовірностями. Для конструювання функцій приналежності перерізу і об’єднання можна використовувати операції:

$$\mu C(x) = \mu A(x) * \mu B(x), \mu D(x) = \mu A(x) + \mu B(x) - \mu C(x).$$

Доповненням нечіткої множини A в X є нечітка множина D з функцією приналежності вигляду:

$$\mu D(x) = 1 – \mu A(x), x \in X.$$

Суму двох нечітких множин A і B у X позначимо через A + B і визначимо як:

$$\mu_{A+B}(x) = \mu_A(x) \vee \mu_B(x),$$

для кожного $x \in X$, де $\vee$ – операція знаходження максимуму, тобто $a \vee b = \max(a, b)$.

Як з практичної, так і з математичної точки зору «жорсткі» операції кращі за «м’які».

Вибір операцій перерізу та об’єднання є одним з основних питань теорії нечітких множин.

Розглянуті операції перерізу і сум мають чітке обґрунтування, вони інтуїтивно зрозумілі і бувають корисними у багатьох додатках. У той же час використовуються більш загальні означення цих операцій. Серед таких означень особливе місце займають t-норми и s-норми (t-конорми), які дають можливість більшої свободи у виборі бажаних властивостей у цих операціях.

При використанні вибраних методів операцій для розв’язування конкретних задач дуже важливою є проблема адекватності операцій на нечітких множинах.

При моделюванні та обґрунтуванні адекватності операцій використовують різноманітні підходи. Серед них можна виділити наступні:

– інтуїтивні, такі, наприклад, як оригінальні методи Л. Заде, в яких прийняті операції обґрунтовуються більш або менш раціональними аргументами;

– аксіоматичні, у яких задається виконання деякого набору раціональних умов, виходячи з яких, використовуючи аналітичні методи, доводиться, що прийняті означення є єдиними, які виконують ці умови;

– експериментальні, в яких створюються деякі психологічні тести для групи індивідів, а потім на основі їх відповідей визначаються мотивовані операції.

Використання перерахованих методів виявляє окремі аспекти адекватності операцій, але не вирішує проблеми у цілому.

Тому при розв’язуванні практичних задач із використанням елементів теорії нечітких множин у зв’язку з відсутністю однозначних результатів теорії необхідно експериментально підібрати той чи інший тип операцій для кожного нового класу задач.

При використанні інтелектуальних технологій обробки даних [4, с. 20] дуже часто функції містять значення з інших записів таблиць, у той же час обробка даних у таблицях виконується по записах. У таких випадках можна використовувати під-

сумкові запити з обчисленнями по групах записів. На основі таких запитів можна створювати звичайні з обробкою підсумкових обчислень та проведенням операції декартового множення між декількома таблицями.

На наш погляд, привабливість функціональних мов викликана такими головними, їм притаманними особливостями:

– функціональні програми незмінно набагато менші, з більшим ступенем абстракції, і доступніше до розуміння порівняно зі своїми аналогами, написаними імперативними мовами;
– функціональні програми придатні для формального аналізу і маніпулювання;
– вони звичайно піддаються реалізації на паралельних машинах.

Аналогічні переваги використання функціонального програмування проявляються і в обробці реляційних баз даних. З точки зору програмування ці особливості привабливі тим, що функціональні програми є свого роду ієрархічними специфікаціями, які часто використовуються у роботах з технології програмування. У той же час внаслідок того, що ми можемо звернутися до звичайного математичного апарату, формальне маніпулювання функціональними програмами виконується відносно просто і при встановленні їх властивостей, і при перетворенні програм у більш ефективну форму.

Таким чином, у результаті проведеного дослідження автор прийшов до таких висновків:

– засоби функціонального програмування дозволяють суттєво розширити можливості SQL-запитів обробки даних;
– функціональне програмування суттєво полегшує процес створення додатків обробки баз даних;
– такі методи обробки даних можуть реалізовувати інтелектуальні методи обчислень (нейронні мережі, «нечітка» логіка).

У наш час техніка функціонального програмування має широке розповсюдження. Її важливість, як засобу, який дозволяє просунутися у розвитку мов високого рівня, буде зростати у практичному користуванні.

Література

1. Филд А. Функциональное программирование / А. Филд, П. Харрисон. – М. : Мир, 1993. – 501 с.
2. Хендерсон П. Функциональное программирование. Применение и реализация / Хендерсон П. ; [пер. с англ.]. – М. : Мир, 1983. – 349 с.: ил.
3. Теория и практика построения баз данных / Д. Крёнке ; [8-е изд.]. – СПб. : Питер, 2003. – 800 с.: ил.
4. Корнеев В.В. Базы данных. Интеллектуальная обработка информации / [Корнеев В.В., Греев А.Ф., Васютин С.В., Райх В.В.]. – М. : Нолидж, 2000. – 352 с.

Олег Кличук

Обработка реляционных баз данных средствами функционального программирования

В статье рассматривается методика использования функционального программирования. Дано описание задач, в которых используются функции, применение которых существенно упрощает проведение анализа данных.

Oleg Klichuk

Relational Database Processing by Means of Functional Programming

The procedure of functional programming usage is considered in the article. some tasks are described where the functions are used which fundamentally simplify the data analysis.

Стаття надійшла до редакції 17.11.2008.